Python Jupyter Notebooks

1 Purpose

Use and navigate *Jupyter* notebooks with *Python*.

2 *Jupyter* Notebook Cell Types

Standard *Jupyter* notebooks can contain a mix of three possible types of cells.

1. **Markdown** - formatted text, formulas, and images
2. **Raw** - non-executable code-like text
3. **Code** - executable *Python* (or other languages) code

Many online notebook interfaces that are based on *Jupyter* may only contain Markdown and Code cells.

2.1 Code Cell Numbering

- `In [ ]:` - input cell before being executed
- `In [1]:` - input cell after being executed
- `Out[1]:` - output cell belonging to `In [1]:`
- Certain output cells do not get numbers, such as those created by printing
- Numbers are not re-used, they just keep counting until the notebook or its kernel is restarted
- The last state of all input and output cells is maintained when a notebook is saved

2.2 Editing and Executing Code Cells

- Edit mode
  - Green border (usually)
  - Text cursor
  - Type commands/text in cells
  - Must be in this mode to type *Python* commands
  - Double-click inside a cell to enter edit mode
  - Press `[return]` with a cell highlighted while in command mode to enter edit mode

- Command mode
  - Blue border (usually)
  - Standard arrow-style pointer
  - Used to manage the notebook's structure and appearance
  - Copy, paste, and delete cells
  - Click outside of a text editing area (left or right of the cell) to enter command mode
  - Execute a cell that is being edited to enter command mode

2.3 Common command mode shortcuts

- **A**: add a new cell above the selected cell
- **B**: add a new cell below the selected cell
- **X**: cut the selected cell
- **M**: change the selected cell to *Markdown*
- **Y**: change the selected cell to Code
- **R**: change the selected cell to Raw

2.4 Executing cells

- All cell types can be executed
  - Code cells execute the *Python* commands
  - *Markdown* cells generate the formatted text
  - Raw cells leave the text "as-is"
- `[shift]+[return]` is used to execute a cell and jump to the next cell or start a new cell
- `[control]+[return]` on Windows (or `[command]+[return]` on a Mac) is used to execute a cell and keep the current cell selected

3 Editing Formatted Text (*Markdown*) and Raw Text Cells

3.1 *Markdown* cells are good for ...

- Instructions, descriptions, explanations, and general commentary
- Linking and displaying image files that are located in the same directory as the notebook file
- Displaying formulas/equations

3.2 Common *Markdown* symbols used for formatting text

- Include `*` immediately before and after text for *italic* text, i.e. `*italics*`
- Include `**` immediately before and after text for **bold** text, i.e. `**bold**`
- Place a `$` before and after special code to generate inline mathematical formulas using LaTeX; i.e. $\beta = \frac{\pi}{4}$
- Use `-` or `*` followed by a space at the beginning of a line to create a bullet
- Use `#` followed by a space at the beginning of a line to make a title (first order heading)
- Use `##` followed by a space at the beginning of a line to make a second order heading
- Use `###` (or more) followed by a space at the beginning of a line for a third (or higher) order heading
- Place `___` (3 underscore characters) at the beginning of a line to make a horizontal dividing line

3.3 Raw cells

- Used less frequently
- Used to show sample code
- Not executable

4 Running Scripts in *Jupyter* Notebooks

- Place the script file in same directory as the *Jupyter* notebook
- Execute `run script.py` in a code cell using the actual script name, i.e. `run hello.py`

```
In[ ]: run hello.py
```

5 Some Special (aka Magic) Commands

- `who` lists module and variable names that are currently in memory
- `whos` is like `who` but it also includes types and value information